

Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software UPM ETSIINF.
Examen de Programación II. Convocatoria ordinaria. 1 de junio de 2018.

Realización: El test se realizará en la hoja de respuesta. Es **importante** que no olvidéis rellenar vuestros datos personales y el código clave de vuestro enunciado. Se pueden utilizar hojas aparte en sucio.

Duración: La duración total del test será de **30 minutos**.

Puntuación: El test se valora sobre **10 puntos**. Las preguntas tipo test pueden tener una única respuesta o varias respuestas, el enunciado lo deja claro. Cada pregunta con una única respuesta respondida correctamente vale 2 puntos, e incorrectamente respondida resta 2/3 puntos. Si en una pregunta con una única respuesta se selecciona más de una respuesta, la pregunta se puntuará con 0 puntos. Para una pregunta con varias respuestas, cada afirmación correcta seleccionada suma 2/no-respuestas-correctas puntos, y cada afirmación incorrecta seleccionada resta 1/no-respuestas-incorrectas puntos. Las preguntas no contestadas suman 0 puntos en cualquier caso.

Calificaciones: Las calificaciones se publicarán en moodle como muy tarde el día **8 de junio de 2018**

Revisión: Las revisiones serán el día **13 de junio de 2018** previa petición por correo electrónico al profesor que se indique por el foro de la asignatura.

Primer Ejercicio

Dada la siguiente clases definida en Java que usa la clase Node vista en clase, y asumiendo que el código compila y ejecuta sin problemas:

```
package ejercicio1;

public class Ejercicio1 {

    public static void main(String[] args) {
        Node<String> head, other;
        head = new Node<String>("coche",
                                new Node<String>("casa"));
        other = new Node<String>("perro",
                                new Node<String>("gato",
                                                  head.next()));
        head.setNext(other);

        other = head;
    }
}
```

```
while (other!=null){
    System.out.print(other.element()+"-");
    other = other.next();
}

}
```

Pregunta 1 (1 punto)

Indique cuál es la salida que se obtendría por consola (sólo una):

- a) coche-perro-gato-casa-
- b) coche-gato-perro-casa-
- c) perro-gato-coche-casa-
- d) coche-casa-perro-gato-

Segundo Ejercicio

Dada las siguientes clases definidas en Java y asumiendo que el código compila y ejecuta sin problemas:

```
public class ClaseA {
    private int datoA;
    protected int datoB;
    public ClaseA(int dato){
        this.datoA = dato+datoB+1;
    }
    public int operar (int info){
        if (info < datoA){//IF
            return info;
        }//IF
        else {//Else
            return datoB;
        }//Else
    }
}

public class ClaseB extends ClaseA {
    private int datoC;
    public ClaseB (int dato1, int dato2){
        super (dato1);
        this.datoC = dato2;
        datoB++;
    }
}
```

```
@Override
public int operar (int info){
    return this.datoC +
           super.datoB + super.operar(info);
}
}
```

Dado el siguiente programa principal ubicado en una clase independiente:

```
public class ClaseC {
    public static void main(String[] args) {
        ClaseB d1 = new ClaseB(10, 20);
        ClaseA d2 = new ClaseA(6);
        ClaseB d3 = new ClaseB(30, 40);

        System.out.print(d1.operar(10) + "_");
        System.out.print(d2.operar(100) + "_");
        System.out.print(d3.operar(1000) + "_");
    }
}
```

Pregunta 1 (1 punto)

Indique cuál es la salida que se obtendría por consola (sólo una):

- a) 66 2 31
- b) 31 0 42
- c) 32 2 44
- d) 33 17 21

Tercer Ejercicio

Dado el siguiente código:

```
package ejercicio3;

public class Vehiculo {

    private String tipoCombustible , marca , modelo ;
    public Vehiculo (String combustible, String marca, String modelo){
        tipoCombustible = combustible;
        this.marca=marca;
        this.modelo=modelo;
    }

    public boolean equals(Object o){
        if (!(o instanceof Vehiculo))
            return false;
        Vehiculo v = (Vehiculo)o;
        return marca.equals(v.marca) && modelo.equals(v.modelo)
            && tipoCombustible.equals(v.tipoCombustible);
    }
}
```

Pregunta 1 (1 punto)

Suponiendo que el código anterior compila correctamente, indique qué afirmación es la correcta con respecto al siguiente código (**sólo una**).

```
public static void main(String[] args) {
    Vehiculo v1 = new Vehiculo("Gasolina", "Chevrolet", "Camaro");
    Vehiculo v2 = new Vehiculo("Gasolina", "DeLorean", "DMC-12");
    Vehiculo v3 = new Vehiculo("Gasolina", "Chevrolet", "Camaro");
    Vehiculo v4 = v1;

    System.out.print(v1.equals(v2)+" "+(v1==v2)+" ");
    System.out.print(v1.equals(v3)+" "+(v1==v3)+" ");
    System.out.print(v1.equals(v4)+" "+(v1==v4)+" ");
}
```

- a) false false true false true false
- b) Todas son falsas
- c) false false true false true true
- d) false false false false true true

Cuarto Ejercicio

Dado el siguiente código, suponiendo que compila correctamente, indique cuál sería el resultado del siguiente código

```
public class EjercicioExcepciones {
    public int metodo(int n)
        throws NoValidoException{
        if (n<=0)
            throw new NoValidoException("Error_");
        int pdv = -1;
        int c = 2;
        while (c<(n/2) && pdv!=-1){
            if (n % c==0)
                pdv = c;
            else
                c++;
        }
        return pdv;
    }
}
```

Pregunta 1 (1 punto)

Suponiendo que el código anterior compila correctamen-

te, indique qué afirmación es la correcta con respecto al siguiente código (**sólo una**).

```
public static void main(String[] args) {
    EjercicioExcepciones e =
        new EjercicioExcepciones();
    try{
        System.out.print(e.metodo(9)+ " ");
        System.out.print(e.metodo(8)+ " ");
        System.out.print(e.metodo(0)+ " ");
        System.out.print(e.metodo(3)+ " ");
    }catch(NoValidoException ex){
        System.out.print(ex.getMessage());
    }
    System.out.print("FIN");
}
```

- a) 3 2 -1 2
- b) -1 3 Error
- c) 3 2 Error FIN
- d) 3 2 Error

Quinto Ejercicio

Dado el código siguiente, suponiendo que compila correctamente, indique cuál sería el resultado del siguiente código

```
public class EjercicioCadenas<E> {
    private Node<E> head;
    public void a(E d){
        if (head==null || head.next()==null){
            head = new Node<E>(d, head);
        }
        else{
            Node<E> aux = head;
            while (aux.next().next()!=null){
                aux = aux.next();
            }
            aux.setNext(new Node<E>(d, aux.next()));
        }
    }
    public String toString(){
        String res = "";
        Node<E> aux = head;
        while (aux!=null){
            res += aux.element() + "_-_-";
            aux = aux.next();
        }
        return res;
    }
}
```

Pregunta 1 (1 punto)

Suponiendo que el código anterior compila correctamente, indique qué afirmación es la correcta con respecto al siguiente código (**sólo una**):

```
public static void main(String[] args){
    EjercicioCadenas<Integer> e = new EjercicioCadenas<Integer>();
    e.a(1);
    e.a(2);
    e.a(3);
    e.a(4);
    e.a(5);
    System.out.println(e);
}
```

- a) 1 - 2 - 3 - 4 - 5 -
- b) 5 - 4 - 3 - 2 - 1 -
- c) 2 - 3 - 1 - 4 - 5 -
- d) 2 - 3 - 4 - 5 - 1 -